

Tecnologia da Informação

Blog dedicado a informações sobre o tema Tecnologia da Informação, a idéia é bloggar sobre todos os assuntos relacionados, desde hardware até Gestão Empresarial, mas meu foco principal sempre será o Informix e o Unix.

PÁGINAS

[Início](#)
[Curriculum Vitae](#)

QUEM SOU EU



Amilcar de Jesus Moreti

 [Seguir](#) 0

Possui o Segundo Grau completo como Técnico em Contabilidade, é Aluno de Graduação na UniSEB Polo de Atibaia, cursando Gestão de Tecnologia da Informação. Realizou aprimoramento na Hewlett Packard do Brasil em sistemas Unix, IBM Brasil em Sistemas Unix e Banco de Dados Informix e na TOTVS na área de Sistemas de Gestão Empresarial se especializando em Tecnologia do Logix, Datasul e Protheus. Tem experiência na área de Administração de Projetos com ênfase em implantações de ERP's. Certificado pela IBM em Banco de Dados Informix "Information Management Informix Technical Professional v3"

[Visualizar meu perfil completo](#)

INFORMIX CENTER 11.5

[Brazilian Informix User Group](#)
[dotProject - Gestão de Projetos](#)
[IMartins - Tudo sobre Informix](#)
[Informix Center v11.50](#)
[Informix Zone](#)
[International Informix User Group](#)
[Patrick Hissa](#)

ARQUIVO DO BLOG

[► 2014 \(4\)](#)
[▼ 2013 \(2\)](#)
[► Outubro \(1\)](#)
[▼ Agosto \(1\)](#)

Configurando SHMMAX e SHMALL para Oracle no Linux

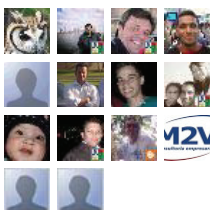
[► 2012 \(21\)](#)
[► 2011 \(48\)](#)
[► 2010 \(47\)](#)
[► 2009 \(22\)](#)

SEGUIDORES

Participar deste site

Google Friend Connect

Membros (14)



Já é um membro? [Fazer login](#)

INSCREVER-SE

TERÇA-FEIRA, 20 DE AGOSTO DE 2013

Configurando SHMMAX e SHMALL para Oracle no Linux

SHMMAX e SHMALL são dois parâmetros chaves de memória compartilhada que impactam diretamente a maneira pela qual a Oracle cria um SGA. Memória compartilhada nada mais é que parte do Unix IPC SYSTEM (Inter Process Communication) mantido pelo KERNEL onde vários processos compartilham um único pedaço de memória para se comunicar uns com os outros.

O ORACLE ao tentar criar uma SGA durante a inicialização do banco de dados, o ORACLE escolhe entre um dos três modelos de gerenciamento de memória:

- um segmento ou
- multi segmentos contíguos
- multi segmentos não contíguos

A adoção de qualquer destes modelos é relacionado ao tamanho da SGA e dos valores definidos para os parâmetros de memória compartilhada no KERNEL do Linux, e o mais importante o SHMMAX.

O que são os parâmetros – SHMMAX E SHMALL?

SHMMAX é o tamanho máximo de um único segmento de memória compartilhada definida em “bytes”.

```
matvir01:~ # cat /proc/sys/kernel/shmmx
2155769856
```

SHMALL é o tamanho total de memória compartilhada disponível em (paginas).

```
matvir01:~ # cat /proc/sys/kernel/shmall
3145728
```

Obs.: é importante notar aqui é o valor de SHMMAX que é definido em “bytes”, e o valor do SHMALL que é definido em “paginas”.

Qual é o valor ideal para SHMALL?

Como SHMALL é o tamanho total de memória compartilhada, deve ser sempre menor do que a memória física do sistema e maior do que a soma das áreas da SGA de todas as bases de dados do ORACLE Server. Uma vez que este valor (soma da SGA) chegar no limite, isto é, o valor do SHMALL, então qualquer tentativa de iniciar um banco de dados novo ou um banco de dados existente com SGA redimensionado, resultara um erro “out of memory”, erro abaixo. Isso ocorre devido não existir mais segmentos de memória compartilhada para o SGA ser alocado.

ORA-27102: out of memory

Linux Error: 28: no space left on device.

Esse erro pode ocorrer por duas razões. O valor de SHMALL não esta definido com um valor ideal ou o servidor já chegou no limite das configurações de memória.

Definir um valor ideal para SHMALL é sempre um trabalho difícil, de paciência, de testes ate chegar no valor ideal. É necessário saber quanto de memória física (excluindo Cache/Swap) existe disponível no sistema e desse valor disponível o quanto você quer deixar para o sistema operacional trabalhar e quanto deixar para o Oracle trabalhar.

Por exemplo, vamos dizer que a memória física de um servidor é de 20GB, e desse valor você quer deixar para o Linux trabalhar uns 8GB e para o ORACLE 12GB, então aqui esta como vamos obter o valor para o SHMALL.

Vamos converter esses 12GB para bytes e dividir pelo tamanho da PAGE SIZE. Lembre-se SHMALL deve ser definido em “paginas” e não em “bytes”

então aqui vai o calculo.

Primeiramente vamos determinar o tamanho do PAGE_SIZE, pode ser feito de duas maneiras. No meu caso é de 4096 e o padrão recomendado e na maioria dos casos esse é o valor padrão dos Linux.

```
matvir01:~ # getconf PAGE_SIZE
4096
```

ou

```
matvir01:~ # cat /proc/sys/kernel/shmmni
4096
```

Agora vamos converter 12GB em bytes e dividir pelo tamanho da pagina.

```
matvir01:~ # echo "( 12 * 1024 * 1024 * 1024 ) / 4096 " | bc -l
3145728.00000000000000000000000000
```

Redefinir SHMALL e carrega-lo dinamicamente no KERNEL.

```
matvir01:~ # echo "3145728" > /proc/sys/kernel/shmall
matvir01:~ # sysctl -p
```

Verificar o efeito

```
matvir01:~ # ipcs -lm
```

```
----- Shared Memory Limits -----
max number of segments = 4096
max seg size (kbytes) = 2105244
max total shared memory (kbytes) = 12582912
```

Tecnologia da Informação: Configurando SHMMAX e SHMALL para Oracle no Linux

min seg size (bytes) = 1

Qual o Valor ideal para SHMMAX?

O ORACLE faz uso de um dos três modelos de gerenciamento de memória para criar o SGA durante a inicialização do banco de dados e faz isso na seguinte sequência. Primeiro a Oracle tenta usar o modelo de um segmento e se falhar ele prossegue com o próximo, que é o modelo multi segmento contíguos e se falhar também ele vai com a última opção, que é o modelo de multi segmentos não contíguos.

Então durante a inicialização, ele procura o parâmetro SHMMAX e compara com o parâmetro de inicialização `*SGA_TARGET`. Se `SHMMAX > *SGA_TARGET`, então o ORACLE vai iniciar com o primeiro modelo, criando o SGA dentro de um único segmento de memória compartilhada.

Porém se a tentativa acima (um segmento) falhar, então o ORACLE continua com a segunda opção - o modelo multi segmentos contíguos. Alocações contíguas, como o nome indica, são um conjunto de segmentos de memória compartilhada, que são contíguas dentro da memória e se ele pode encontrar um conjunto de tais segmentos, em seguida, todo SGA é criado para caber dentro deste conjunto.

Mas se não pode encontrar um conjunto de atribuições contíguas então a última opção é escolhida – alocação multi segmentos não contíguos e neste caso o ORACLE pega segmentos de memórias livres fragmentadas entre espaços utilizados.

Então digamos que se nós sabemos o tamanho máximo de SGA de qualquer banco de dados no servidor permaneça abaixo de 1GB, então podemos definir SHMMAX para 1GB. Mas dizer que temos tamanhos de SGA diferentes para os bancos de dados espalhados entre 512MB a 2GB, então defina SHMMAX para 2GB e assim por diante.

Alterando o valor de SHMMAX

```
matvir01:/etc/sysconfig # echo "8589934592" > /proc/sys/kernel/shmmax
```

```
matvir01:/etc/sysconfig # sysctl -p
```

Os documentos de instalação do ORACLE 11g recomenda o valor de SHMMAX como “4GB – 1 byte” ou metade do tamanho da memória física o que for menor. Eu acredito que 4GB esta relacionado com a limitação de sistemas de 32bits (x86) onde o espaço de endereço virtual de um processo de usuário pode ser menor do que 4GB. Como não há nenhuma limitação para sistemas de 64 bits (x86_64), você pode definir SGA maior que 4GB. Mas a ideia aqui é fazer com que o ORACLE utilize o modelo eficiente de um único segmento e para isso o SHMMAX deve ficar maior que o tamanho da SGA.

Postado por [Amilcar de Jesus Moreti](#) às 14:53



+1

Recomende isto no Google

Reações:

Interessante (1)

Otimo (1)

Bom (0)

Regular (0)

Ruim (0)

Nenhum comentário:**Postar um comentário**

Digite seu comentário...

Comentar como:

Conta do Gooç ▼

Publicar

Visualizar

Links para esta postagem

[Criar um link](#)

[Postagem mais recente](#)

[Início](#)

[Postagem mais antiga](#)

Assinar: [Postar comentários \(Atom\)](#)

Modelo Picture Window. Imagens de modelo por [enot-poloskun](#). Tecnologia do [Blogger](#).